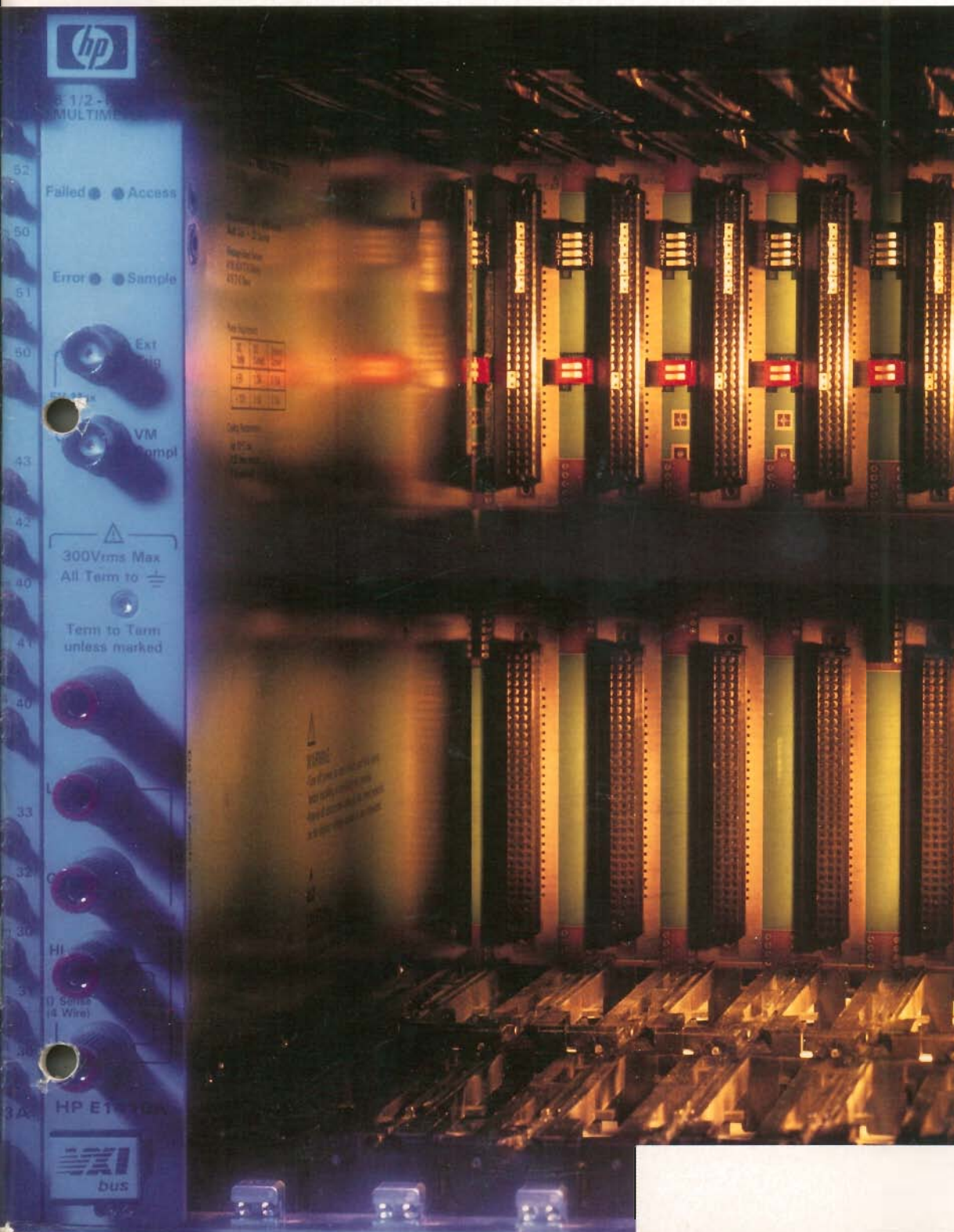


HEWLETT-PACKARD JOURNAL

April 1992



A High-Performance Measurement Coprocessor for Personal Computers

This plug-in card brings test and measurement coprocessing power to ISA (Industry Standard Architecture) personal computers with greater calculation speed and better HP-IB performance than its predecessor. It also has DMA capability.

by Michael P. Moore and Eric N. Gullerud

The HP 82324A high-performance measurement coprocessor is a plug-in card for HP Vectra and compatible computers that turns an ordinary PC into a multiprocessing test and measurement workstation. The coprocessor is programmed within the DOS environment using HP BASIC, a de facto standard test and measurement programming language.

The HP 82324A high-performance measurement coprocessor is designed to meet customer needs for higher calculation speed and better HP-IB performance than its predecessor, as well as DMA for better overall system performance. To minimize duplicated effort and maximize reliability, the design of the measurement coprocessor is leveraged from the HP 9000 Model 332 computer. The Model 332 was chosen because of its low cost, high performance, and potential for fitting onto a single full-size PC I/O card.

Hardware Architecture

Fig. 1 is a block diagram of the high-performance measurement coprocessor. At the heart of the design is the

16-MHz MC68030 CPU with its integral memory management unit. The MC68882 floating-point coprocessor can be installed as a socketed option. A custom DMA controller allows rapid transfers of data between memory and devices. Plug-in RAM boards, similar to the Model 332 RAM boards, are available in 1M-byte and 4M-byte sizes. One or two RAM boards can be plugged into the main board, allowing RAM configurations of 1M, 2M, 4M, 5M, and 8M bytes. Built-in HP DIO input/output bus circuitry provides a connection to companion HP GPIO and HP SRM (shared resource manager) interface cards, and the on-board HP-IB interface allows direct connection to HP-IB (IEEE 488, IEC 625) instruments and devices.

Features eliminated from the Model 332 design include the display circuitry, keyboard controller, timer, speaker, and serial I/O. Those functions are performed by PC resources through an emulation process discussed later. Space constraints required the elimination of the memory parity circuitry. The boot ROMs were replaced by a scheme that uses the CPU's memory management unit to remap RAM downloaded from the PC into the boot ROM

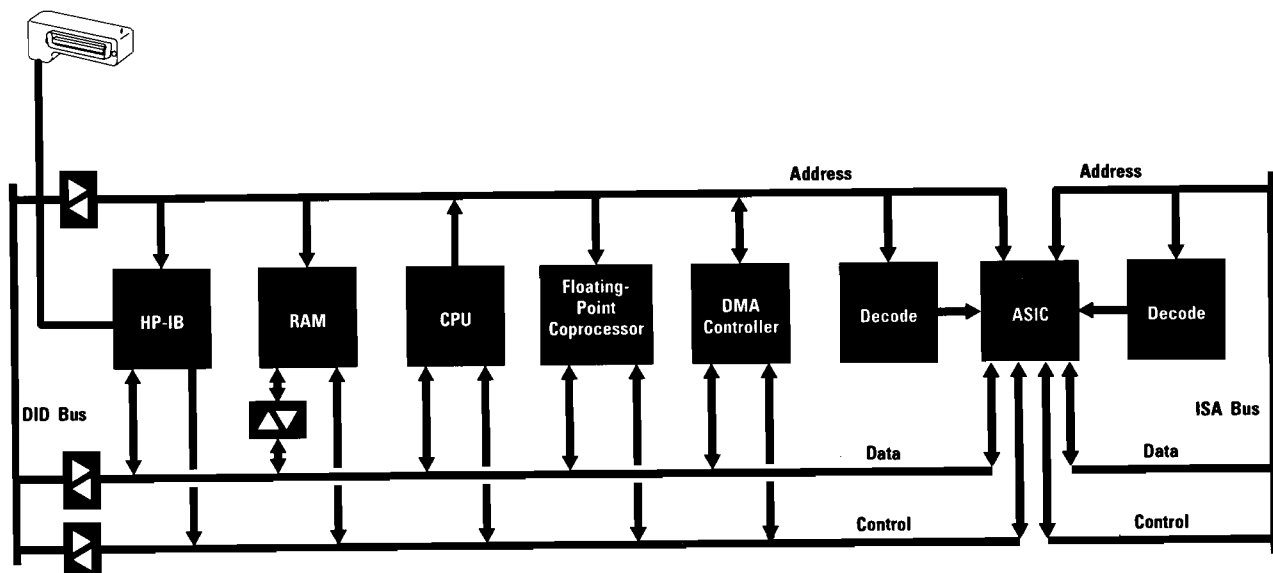


Fig. 1. Block diagram of the HP 82324A high-performance measurement coprocessor.

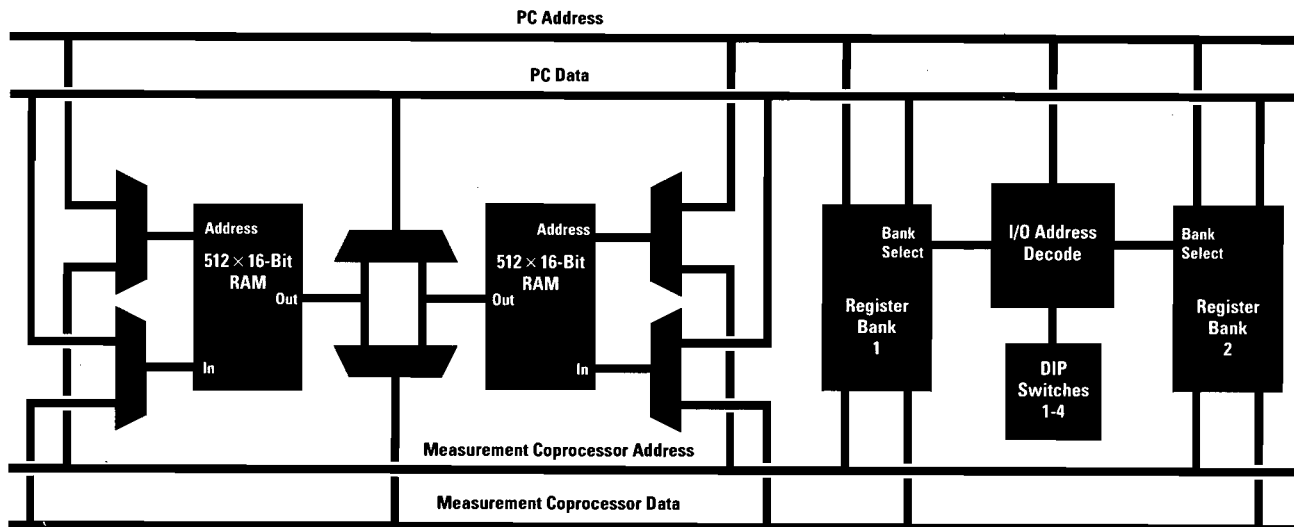


Fig. 2. Block diagram of the application-specific integrated circuit in the measurement coprocessor.

address space. Added to the Model 332 design is an ASIC (application-specific integrated circuit) that provides an enhanced interface to the PC backplane. It includes the same interface mechanism used by its predecessor (detailed in the section below on device emulation), as well as a new memory-based mechanism that allows a higher-bandwidth data path to the PC. Fig. 2 is a block diagram of the ASIC.

PC Backplane Interface

The interface to the PC backplane consists of two banks of eight 8-bit I/O registers, two 1024-byte memory buffers, and three interrupt sources. The base address of the I/O registers is configurable by a DIP switch at the top of the main board. The base address of the memory buffer and the interrupt level are both configurable through the I/O registers. The user can reconfigure the card without having to remove it from the computer and without turning off the computer. Four switches are used to select the base address of the I/O registers. Two of the switches select one of four address ranges within the PC's standard 10-bit I/O address range. These addresses are 250h, 280h, 330h, and 390h. The other two switches select one of four "alias" address ranges outside of the PC's 10-bit I/O address range, making the base address a 12-bit value. In other words, if the first two switches select a base address of 250h, then the other two switches further qualify the address as one of 250h, 650h, A50h, or E50h. This addressing scheme allows up to four HP 82324A cards to be mapped into a single 8-bit register bank, conserving scarce PC I/O resources. Finally, the second bank of registers is selected by the thirteenth address bit. Thus, if the first eight registers are at 250h through 257h, the second eight are at 1250h through 1257h. Again, this scheme conserves PC I/O resources.

One of the I/O registers selects one of eight possible PC interrupt lines and one of eight base addresses for the memory buffer. It also selects either the 8-bit or the 16-bit access mode of the memory buffer or disables it completely.

Another I/O register is used to generate interrupts to the 68030 from the PC. This enables the PC software to emulate devices that generate interrupts, such as the keyboard controller. This register is also used to enable or disable the three interrupt sources that generate PC interrupts. These interrupt sources are a 10-millisecond periodic interrupt, a PC mailbox interrupt, and an address match interrupt.

The 10-millisecond interrupt source is used as the time base for the keyboard controller emulation. In an HP 9000 Series 300 computer, the keyboard controller contains timers used for keeping the time and date, for timeouts, for periodic event generation, and for delays. These timers all have a resolution of 10 milliseconds. Since the PC's periodic system interrupt occurs only approximately every 55 milliseconds, the 10-millisecond interrupt source on the measurement coprocessor was necessary. The mailbox interrupt occurs when the 68030 sets the PC mailbox flag. The purpose of this flag is discussed later. The address match interrupt occurs when a bus cycle initiated by the 68030 is within an address range reserved for device emulation. This interrupt source can be used to implement interrupt-driven device emulation instead of polled device emulation.

Device Emulation

When the 68030 (or some other DIO bus master, such as the DMA controller) generates a bus cycle within a certain range of addresses, the ASIC freezes the bus cycle and waits for PC software to complete it. If the 68030 is performing a write operation, the PC software can read the value and release the bus cycle. If the 68030 is doing a read, the PC software can place a value on the data bus and release the bus cycle. To the 68030, nothing different is happening from writing to an actual hardware device, such as the built-in HP-IB interface, except that the bus cycles take much longer to complete. The PC has complete control over the termination of one of these "trapped" cycles, and can take its time determining whether to terminate the bus cycle normally or abort it with a bus error.

(continued on page 113)

HP Computer Museum
www.hpmuseum.net

For research and education purposes only.

Measurement Coprocessor ASIC

The custom application-specific integrated circuit (ASIC) in the HP 82324A measurement coprocessor design contains the interface between the Industry Standard Architecture (ISA) backplane bus of the PC and the CPU bus of the measurement coprocessor. The interface to the PC supports either 8-bit or 16-bit access modes while the measurement coprocessor interface is fixed at 16 bits.

The ASIC has outputs directly connected to six interrupt lines on the measurement coprocessor interface so that PC software can generate multiple interrupts. On the PC, each interrupting device must have a dedicated interrupt line. At boot time, three software-controlled outputs select one of eight possible interrupts to connect to the ISA bus by external circuitry.

Sixteen registers in the ASIC are addressable from the PC interface and six are addressable from the measurement coprocessor interface. In the PC interface the registers provide a view and limited control of the state of the measurement coprocessor bus. Data can be read from or written to the data bus during trapped

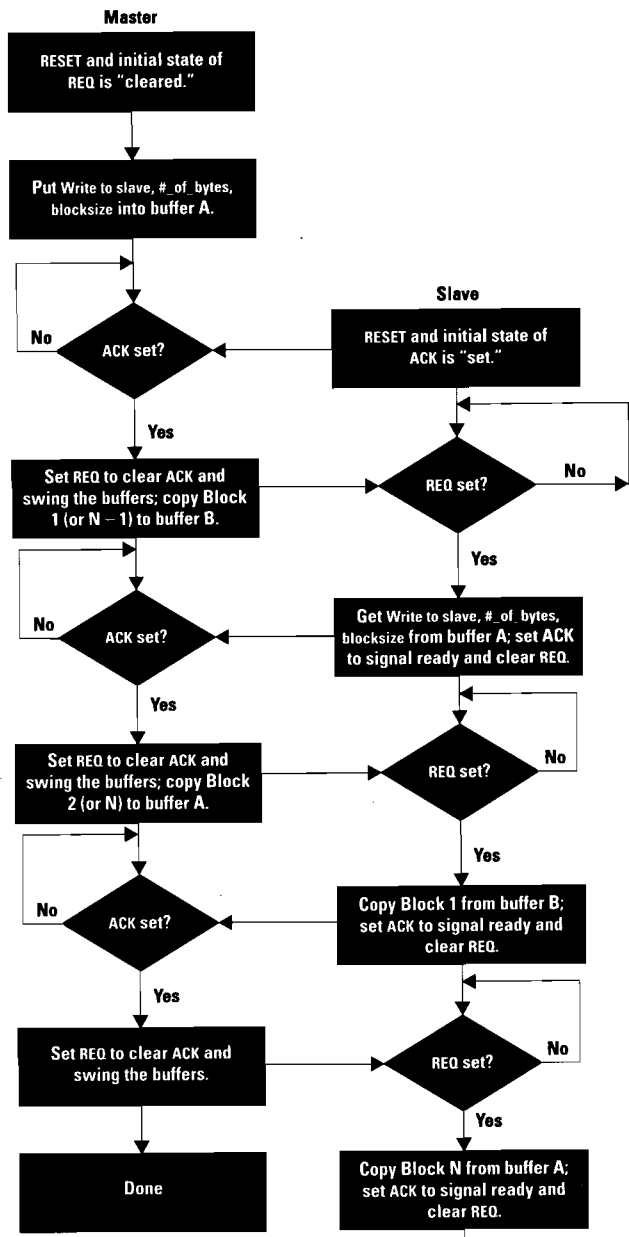


Fig. 1. Data transfer from master to slave.

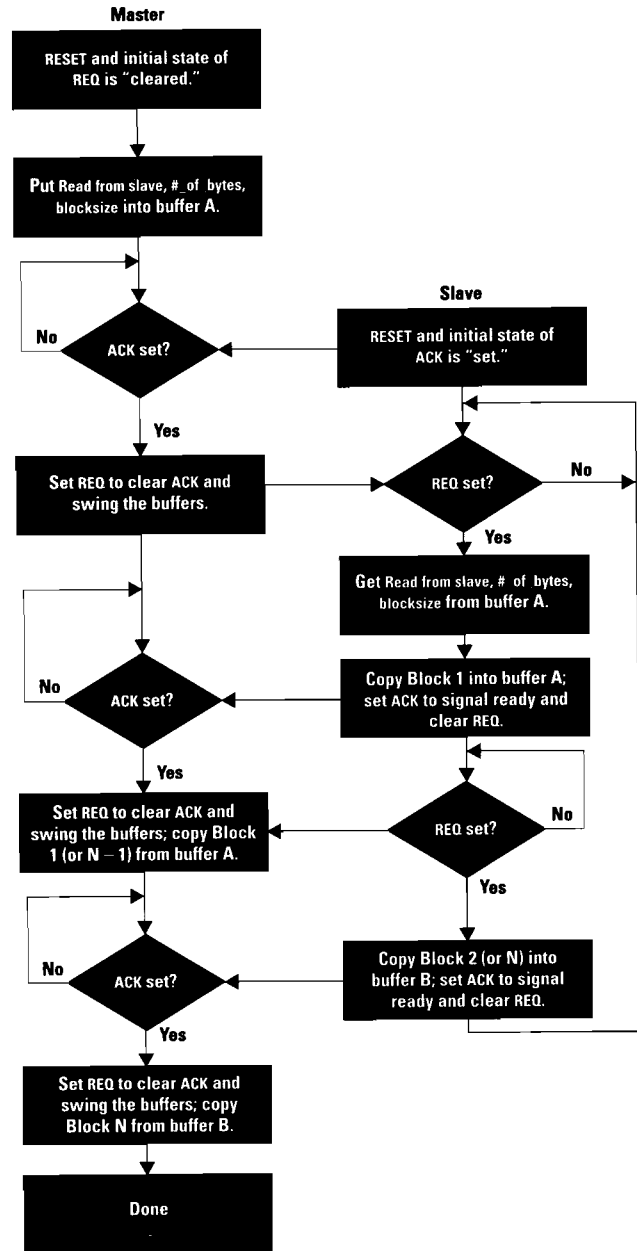


Fig. 2. Data transfer from slave to master.

accesses, and the cycle is then ended under register control. (For a description of trapped accesses, see "Device Emulation" in the accompanying article). The PC can also assert any of the six shared priority-level interrupts on the measurement coprocessor. A typical use of these interrupts is as follows. The PC receives a keystroke for the measurement coprocessor and asserts an interrupt as if it were the keyboard controller. The measurement coprocessor responds to the interrupt by addressing the keyboard controller register to read the keycode. The read cycle is trapped and recognized by the PC, which writes the keycode to the measurement coprocessor data bus and asserts the signal to end the read cycle.

There are three types of interrupts for the PC, which can be enabled separately. These are generated by a 10-millisecond timer, a semaphore flag in the measurement coprocessor interface, and address matches. Matching addresses are exter-

nally decoded and categorized by three inputs to the ASIC: alpha accesses, graphics accesses, and all others. The PC can view these categories to determine what action needs to be taken.

The DMA controller on the measurement coprocessor relies on external resources to perform byte folding, that is, duplicating data from the low byte of the bus to the high byte when necessary for transfers involving 8-bit devices. Byte folding is handled within the ASIC on demand from the DMA controller.

Byte addressing by the Motorola CPU of the measurement coprocessor is different from byte addressing by the Intel CPUs used in PCs. The least-significant byte of a word has an odd address for Motorola and an even address for Intel. For files as well as for some data types, it is necessary to swap bytes whenever they are stored on the PC. The ASIC can perform byte swapping in hardware if enabled via a register in the PC interface.

Two banks of 512×16 bits of RAM are included in the ASIC to facilitate fast block transfers between the measurement coprocessor and the PC. The two RAM banks are configured as a "swing buffer" called the HyperChannel. Each bank can be accessed from either the PC interface or the measurement coprocessor interface

The PC can handle the emulation of devices either by waiting for an address match interrupt, as mentioned above, or by polling the status of the 68030's bus. The polling method allows the PC to emulate devices faster than an interrupt-based method, but emulation software must be constantly polling the interface instead of waiting for an event.

While the trapped address mechanism simplifies the PC software and initially eliminated the need to modify HP BASIC software, it has two major drawbacks. First, the rate of data transfer between the PC and the measurement coprocessor is relatively slow. This often makes the transfer of large blocks of data the bottleneck in PC/measurement coprocessor performance. Second, no interrupts, even nonmaskable, can get through to the 68030 when it is in the trapped address state. This makes interrupt-driven I/O without hardware handshaking unreliable at best, and impossible in some situations.

Improved Interprocessor Communication

To circumvent these drawbacks, the PC interface has a secondary communication channel called the HyperChannel. This memory-based mechanism consists of two 1024-byte buffers and three handshaking flags. When one of the buffers is accessible to the PC, the other is accessible to the 68030, and vice versa. Whichever processor is designated the channel master controls the swapping of the buffers when both processors are ready. Since the buffers are being accessed simultaneously, the sustained data transfer rate is as fast as the slower of the two processors. In addition to the performance advantage, this protocol allows the measurement coprocessor and the PC to transact business while both processors remain completely interruptible by other tasks. See "Measurement Coprocessor ASIC," page 112, for a detailed description of the buffer protocol.

A third communication mechanism has been added to make the interface even more flexible. This communication path consists of two mailbox flags, one for the PC and one for the measurement coprocessor. Both mailbox

but not from both simultaneously. While the PC is accessing one bank, the measurement coprocessor can be independently accessing the other. When both the PC and the measurement coprocessor have finished their respective accesses, the buffers can be swapped or "swung" between the two interfaces. This allows one interface to be continuously filling buffers while the other interface is continuously emptying them in a fully simultaneous process. In this way, a transfer can be accomplished at the full speed of the slower interface.

An identical set of registers is provided for each interface to synchronize transfers using the HyperChannel. Only one interface can initiate a transfer and is therefore considered the owner or master of the HyperChannel, while the other interface is considered the slave. Either interface can be master, but ownership can only be relinquished by the current master, not preemptively seized by the slave. Identical sets of four single-bit write registers and one read register exist in each interface to control the HyperChannel. The read registers include four bits indicating the state of each of the write registers. The write registers allow the assertion of the signals request (REQ), acknowledge (ACK), error (ERR), and change master. The master asserts REQ, which in turn clears ACK and swings the buffers. The slave asserts ACK, which in turn clears REQ. Figs. 1 and 2 show the protocol for transfers between master and slave.

flags can be read by either processor, but only the mailbox flag owned by a processor can be set or cleared by it. When the measurement coprocessor sets its mailbox flag, an interrupt can be generated to the PC. The two processors can then synchronize operations by waiting for both mailbox flags to be set, clearing their mailbox flags, and waiting for both mailbox flags to clear. This protocol allows the measurement coprocessor to generate a PC interrupt without having to be frozen in the trapped address state.

Software Architecture

There are two types of software for the measurement coprocessor: software that runs on the measurement coprocessor itself, and software that runs on the host PC. Both groups of software work cooperatively and concurrently to exploit the capabilities of the measurement coprocessor architecture.

The software that runs on the measurement coprocessor is HP's version of the BASIC language, which has been prevalent in the instrument control world for many years. Because the hardware architecture is leveraged from the HP 9000 Series 300 product line, existing HP BASIC programs can run on the measurement coprocessor, usually without modification. Also, the reference manuals are identical to those shipped with HP BASIC for workstations. Even the measurement coprocessor version of the HP BASIC system is compiled from the same source code that is used to generate the workstation version. (For the history of the development of HP BASIC for the PC, see "Measurement Coprocessor History," page 114.)

In addition to running the HP BASIC system, the measurement coprocessor emulates the system boot ROM in RAM, eliminating the requirement for ROMs on the measurement coprocessor. To achieve ROM-less operation, the PC loads a small boot loader program into the measurement coprocessor's RAM. After testing system RAM, this boot loader program copies the boot ROM image from the PC to the bottom of system RAM, initializes the 68030's memory management unit to map logical

Measurement Coprocessor History

In the early 1980s, the HP 9000 Series 200 computer was HP's premier instrument controller. With HP's version of the BASIC language, the Series 200 made automating part or all of the test and measurement task much easier. HP BASIC became a de facto standard test and measurement language.

The early 1980s also saw the introduction of the IBM personal computer (PC). Within a few years, the IBM PC and compatible machines replaced the Series 200 and 300 computers as instrument controllers for a large portion of the test and measurement market. However, there was no implementation of BASIC for the PC that offered the rich feature set HP BASIC customers had come to expect. HP's answer to this problem was the BASIC language processor—a plug-in card for the PC that temporarily turned the PC into an HP 9000 Series 200 instrument controller.

Two basic strategies for porting the HP BASIC environment to the PC platform were initially considered. One approach was to rewrite the HP BASIC software so that it would run directly in the DOS environment. Because HP BASIC was heavily optimized for the HP 9000 hardware environment, this approach required the commitment of many resources and involved unknown risks.

The other approach was to "port" enough of the HP 9000 hardware to the PC platform so that HP BASIC would run indirectly in the DOS environment. In other words, HP BASIC programs would run on a coprocessor card, while the PC emulated HP 9000 hardware that wasn't on the coprocessor card. While this approach eased the software effort, it required the customer to purchase additional hardware for the PC.

After weighing and considering the two approaches, HP chose the latter. This approach eliminated many of the technical hurdles involved in moving the HP BASIC programming environment to a completely different operating system while retaining compatibility with the original environment.

In the middle of 1987, the BASIC language processor was introduced. It featured an 8-MHz 68000 CPU, an HP-IB interface, boot ROMs, half a megabyte of RAM, an HP DIO bus interface, and special circuitry to interface to the PC's backplane. The heart of the interface circuitry was a mechanism that allowed a program running on the PC to emulate hardware not present on the BASIC language processor. Daughter boards were available to add more RAM or ROM. Sister boards were available to provide HP GPIO and HP SRM (shared resource manager) interfaces. The same HP BASIC software that ran on HP 9000 Series 200 computers ran unmodified on the BASIC language processor with an emulator program running on the PC. Even the boot ROMs were Series 200 boot ROMs.

references to boot ROM address space to the image in RAM, and transfers control to the RESET exception vector in the boot ROM image. From that point on, the boot ROM and the HP BASIC system operate as if a boot ROM were actually present.

Host PC Software

The software that runs on the host PC is grouped into four parts. The system software configuration part is handled by the CONF.EXE program. The boot process and workstation emulation parts are coordinated by the BASIC.EXE program. DOS/HP BASIC communication is handled partly by the BASIC.EXE program and the HPBLP.SYS device driver, and partly by the POP.COM.COM program.

The CONF.EXE program allows the user to alter the software configuration interactively so that the PC's keyboard, serial ports, and HP-IB cards can be remapped as desired. It also allows the user to specify how to map the PC's disk drives to appear as emulated LIF (HP logical interchange format) disk drives. Configuration

The emulator software that ran on the PC had three major functions. First, it mapped the I/O resources of the PC onto Series 200 hardware. Second, the emulator software allowed HP BASIC programs to access the DOS operating system by sending data to and reading from an imaginary GPIO interface. Standard DOS commands, commercial applications software, and custom programs could all be invoked from the HP BASIC environment. Third, the emulator software allowed a limited form of background operation. The emulator could be placed in the background, allowing the BASIC language processor to execute the current HP BASIC program while a completely different application, such as an editor or spreadsheet program, ran on the PC.

Because of the hardware emulation scheme, most programs that ran on Series 200 computers could be run on the BASIC language processor with little or no modification. Even programs that directly accessed hardware (such as the graphics frame buffer) could run unmodified. However, this degree of compatibility had a price: severe performance degradation. Improving performance quickly became a priority.

The next major revision of the software included four major architectural changes. First, the boot ROMs were rewritten to speed up the boot process by an order of magnitude. Second, a new HP BASIC binary program was written that implemented the DOS file system. This provided a way for HP BASIC programs to access DOS files directly, without the clumsy emulation scheme described above. Third, internal alpha, graphics, and DOS file system operations were reorganized on a transaction basis rather than a hardware emulation basis. In other words, the HP BASIC binaries that implemented alpha and graphics video operations and the new DOS file system binary program generated transactions instead of register-level accesses. The emulator program on the PC was rewritten to service these transactions as well as its hardware emulation tasks, and the background mode of operation was enhanced to allow DOS file system operations in the background, enabling an HP BASIC program to log data to disk while another DOS application was running. Fourth, a mechanism for a background HP BASIC program to communicate with a foreground DOS application (such as a spreadsheet) was added. Because of these expanded foreground/background capabilities, the BASIC language processor was renamed the measurement coprocessor. It represented a great improvement in boot performance, graphics operations, and mass storage operations, and provided greater versatility in the DOS environment.

However, while the new software emulator made significant improvements in some areas of performance, the remaining issues could only be addressed by upgrading the hardware. These issues included increased computation speed, HP-IB I/O throughput, and DMA capabilities. From this effort came the high-performance measurement coprocessor described in the accompanying article.

information is stored in a file that is used by the emulation programs described below.

Boot Process

The BASIC.EXE program is the main control program for the boot and workstation emulation tasks. When BASIC.EXE is run, it first checks to see if the measurement coprocessor has been booted. If it has not, it starts the boot process. The boot process is handled by two separate programs, both of which are run from BASIC.EXE.

The first program, B0.EXE, is responsible for starting the measurement coprocessor and loading the boot ROM image into it. The second program, B2.EXE, is responsible for managing the boot process, which is an emulation of the HP workstation boot process. Both programs request hardware configuration information from a device driver (HPBLP.SYS), which is installed by CONFIG.SYS when DOS boots up.

When B0.EXE begins execution, it requests the hardware configuration information for the measurement coprocessor being booted from the HPBLP.SYS device driver and

verifies that there are no hardware configuration conflicts with DOS or other applications. It then resets the measurement coprocessor, which leaves the 68030 frozen in the trapped address state trying to fetch the stack pointer from address \$00000000. B0.EXE then copies a boot loader program to the HyperChannel buffer, and emulates boot ROM accesses long enough to cause the 68030 to exchange buffers and run the program in the buffer. Once the boot loader program is running, B0.EXE copies the actual boot ROM image to the measurement coprocessor over the HyperChannel.

The B2.EXE program handles the boot process for the measurement coprocessor. This program cooperates with the boot code loaded by the B0.EXE program to simulate the Series 300 boot process. Systems can be booted from a variety of sources, including the optional HP SRM interface or external HFS (HP hierarchical file system) disks. To the user, the PC behaves like an HP 9000 Model 332 computer booting up.

Workstation Emulation

After successfully booting the measurement coprocessor, the BASIC.EXE control program, in conjunction with the B3.EXE program and the HPBLP.SYS device driver, emulates the portions of the workstation hardware not present on the measurement coprocessor. The emulation services provided include the emulation of the workstation's display, keyboard, beeper, and mouse, mapping of the PC HP-IB, serial, and printer ports to emulated workstation I/O cards, and mapping of PC disk drives to workstation LIF and DOS formatted disk drives. A mechanism for running DOS commands from the measurement coprocessor is also provided.

The workstation emulation task can be run either in the foreground or in the background. In foreground mode, the PC behaves as if it were an HP 9000 Model 332 computer running HP BASIC. In the background mode, the user runs DOS applications while the measurement coprocessor concurrently runs an HP BASIC program without a visible display or keyboard.

The BASIC.EXE program is responsible for emulation services common to both foreground and background modes. These services include access to the DOS file system, emulation of the timekeeping and beeper parts of the workstation keyboard controller, and the buffering of alpha video. When HP BASIC is placed in the background mode, BASIC.EXE remains resident in DOS memory while other DOS applications execute.

The full workstation emulation is provided by the B3.EXE program, which is started by BASIC.EXE. B3.EXE provides all emulation services not provided by BASIC.EXE. Communication between BASIC.EXE and B3.EXE is facilitated by code and data resident in the HPBLP.SYS device driver. The common data includes state variables and procedure entry points, and the common code includes interrupt service routines.

Because B3.EXE is not executing during background operation, all processing on the measurement coprocessor stops when an HP BASIC program tries to access emulated services available only in foreground mode. Process-

ing resumes when the measurement coprocessor is switched back into foreground mode.

DOS/HP BASIC Communication

Because the measurement coprocessor runs concurrently with the host PC, there is a potential for dividing the test and measurement task between the processors. To take advantage of this potential, the measurement coprocessor must be able to communicate with DOS and DOS applications. The measurement coprocessor software supports this interprocessor communication with three basic mechanisms: shared file access, MultiCom, and PopCom.

When the measurement coprocessor is in the background mode, an HP BASIC program can write to or read from a DOS file, even if a foreground application is using the file system. While DOS is not a multitasking operating system, it is possible for a collection of interrupt service routines to give a background application access to the DOS file system when the foreground application is not using it. The background service routines in BASIC.EXE, in conjunction with the interrupt service routines in the HPBLP.SYS device driver, create this illusion of concurrent use of the DOS file system, which is the basis for this form of interprocessor communication. While this scheme can be used by any application that can access a DOS file, it is relatively slow.

The MultiCom mechanism for DOS/HP BASIC communication uses the existing I/O capabilities of DOS and HP BASIC compiled subroutine libraries to allow HP BASIC programs to send messages to and receive messages from DOS applications. With MultiCom, an HP BASIC program running in the background can generate keypresses to the DOS application running in the foreground by calling one of several compiled subroutines. Conversely, a DOS application running in the foreground can send messages to the HP BASIC program running in the background by writing to a measurement coprocessor device file (similar to the .PRN file for a printer). To illustrate this mechanism, assume that the user wants to run a test, then integrate the data into a spreadsheet to generate a printed report. The user programs the measurement coprocessor to handle the setup, initiation, and data gathering parts of the test, using MultiCom to receive parameters from and send data to the spreadsheet program. The user programs the spreadsheet program to start the test on the measurement coprocessor, receive the data, and graph the results on a form. The user uses HP BASIC for the test task and the spreadsheet program for the presentation task instead of trying to do both tasks in one environment or the other.

Another use of the MultiCom mechanism is to support multiple measurement coprocessor configurations. With MultiCom, a DOS application can communicate with up to three measurement coprocessors in a PC. In addition, one measurement coprocessor can communicate with another independently of the DOS application running in the foreground. In this manner, powerful multiprocessing test systems can be configured and controlled in one host PC.

Finally, a user may simply want to start a test running on the measurement coprocessor, then go do something else

in DOS, such as reading electronic mail or writing a document. However, if the user wants to be notified when the test finishes, or just wants to see how far along it is, it would be annoying to have to stop working, start up HP BASIC, type a couple of commands, exit HP BASIC, and restart the DOS application. The PopCom mechanism supports a kind of "pop-up" interface to HP BASIC running in the background. With PopCom installed, an HP BASIC program in the background can cause a dialog window to pop up over the foreground DOS application, temporarily taking control of the PC. After the user enters a response, the pop-up window disappears, and the foreground application continues execution. This type of pop-up interface is already popular with PC users, and fits naturally into the DOS environment.

Acknowledgments

The original HP 9000 Model 332 CPU board design was done by Mark Anderson of the former Fort Collins Systems Division. Carl Thomsen did the ASIC design before transferring to the Vancouver Division before the end of the project. Greg Parets joined the project toward the end and was instrumental in the final stages of development and release to manufacturing. The software development team for the HP 82324A's predecessor, the HP 82300A basic language processor, was managed by Andy Rood and included Terry Leeper (project leader), Everett Kaser, Leon Nelson, Roberto Orozco, and Tony Yokoyama. Roberto also took over as project manager from Andy during the second major revision of the BLP software. David Kepler managed the hardware design and software QA of both the HP 82300A and the HP 82324A. Support from the RMB/WS team at MSO was invaluable.

HEWLETT-PACKARD
JOURNAL

